

Computational Symbiosis

Methods That Meld Mind and Machine

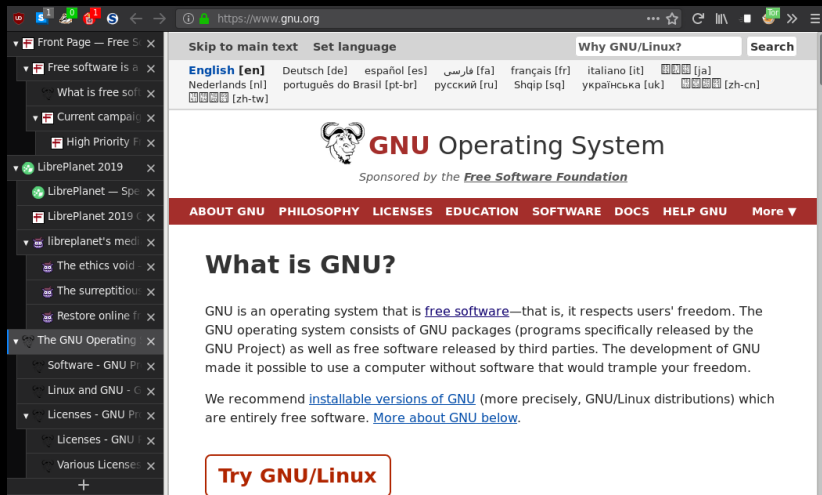
Mike Gerwitz

LibrePlanet 2019

Choreographed Workflows

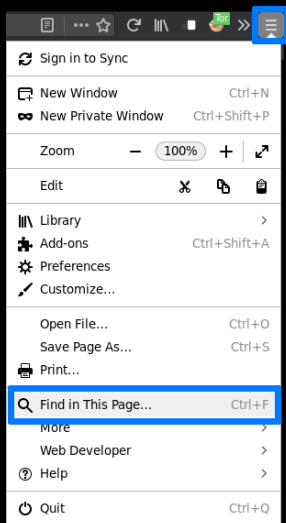
Practical Freedom

Example: Web Browser



The screenshot shows a web browser window with the address bar displaying `https://www.gnu.org`. The browser's tab bar on the left contains several tabs, with the active one being "The GNU Operating System". The main content area of the browser displays the GNU Operating System website. At the top, there are navigation links for "Skip to main text" and "Set language", along with a search box containing the text "Why GNU/Linux?". Below the search box, there is a list of language options: English [en], Deutsch [de], español [es], فارسی [fa], français [fr], italiano [it], 日本語 [ja], Nederlands [nl], português do Brasil [pt-br], русский [ru], Shqip [sq], українська [uk], and 中文 [zh-cn]. The main heading of the page is "GNU Operating System", accompanied by the GNU logo (a ram's head) and the text "Sponsored by the Free Software Foundation". A dark red navigation bar contains the following links: ABOUT GNU, PHILOSOPHY, LICENSES, EDUCATION, SOFTWARE, DOCS, HELP GNU, and More. The main content area features a large heading "What is GNU?" followed by a paragraph explaining that GNU is an operating system that respects users' freedom and consists of GNU packages and free software from third parties. Below this, another paragraph recommends installing GNU/Linux distributions, with a link to "More about GNU below". At the bottom of the page, there is a prominent red button with the text "Try GNU/Linux".

Finding Text (Mouse-Driven GUI Interaction)

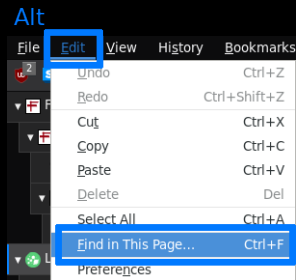
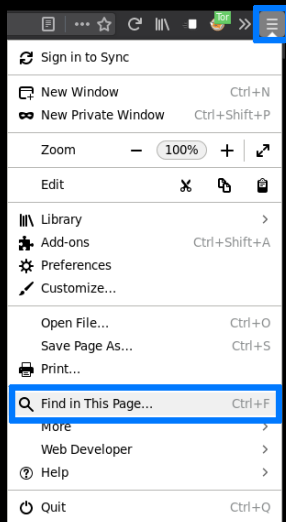


LibrePlanet is an annual conference hosted by the **Free Software** Foundation for **free software** enthusiasts and anyone who cares about the intersection of technology and social justice. LibrePlanet brings together software developers, law and policy experts, activists, students, and computer users to learn skills, celebrate **free software** accomplishments, and face challenges to software freedom. Newcomers are always welcome, and LibrePlanet 2019 will feature programming for all ages and experience levels.

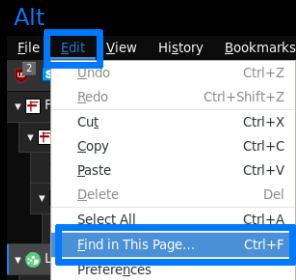
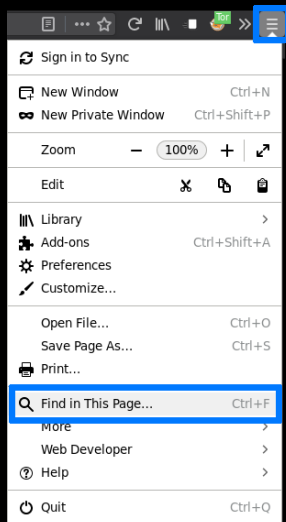
LibrePlanet 2019's theme is "**Trailblazing Free Software**." In 1983, the **free software** movement was born with the announcement of the GNU Project. FSF founder Richard Stallman saw the dangers of proprietary code from the beginning; when code was kept secret from

free software ^ v Highlight All Match Case Whole Words 1 of 11 matches

GUIs Change Over Time



GUIs Change Over Time



Ctrl+F

Muscle Memory

Visual \Rightarrow Tactile

Research Task:

Given a list of webpage URLs

find all that *do not* contain “free software”

Executing the Research Task

Mouse

- 1 Click '+' for each new tab, enter URL
- 2 Menu → Find in This Page
- 3 Type "free software"
- 4 If found, go to #9
- 5 If *not* found, highlight URL, right-click, copy
- 6 Click on text editor
- 7 Right-click, paste URL, hit RET for newline
- 8 Click on web browser
- 9 Click 'X' on tab, go to #2

Executing the Research Task

Mouse

- 1 Click '+' for each new tab, enter URL
- 2 Menu → Find in This Page
- 3 Type "free software"
- 4 If found, go to #9
- 5 If *not* found, highlight URL, right-click, copy
- 6 Click on text editor
- 7 Right-click, paste URL, hit RET for newline
- 8 Click on web browser
- 9 Click 'X' on tab, go to #2

Keyboard

- 1 `Ctrl+T` for each new tab, enter URL
- 2 `Ctrl+F` to find
- 3 Type "free software"
- 4 If found, go to #9
- 5 If *not* found, `Ctrl+L` `Ctrl+C` to copy URL
- 6 `Alt+Tab` to text editor
- 7 `Ctrl+V` `RET` to paste URL and add newline
- 8 `Alt+Tab` to web browser
- 9 `Ctrl+W` to close tab, go to #2

Same Keybindings Across (Most) GUIs!

Browser, Editor, Window Manager, OS, ...

Macro-Like

Ctrl+T ‘‘https://...’’ <N times>

Ctrl+F ‘‘free software’’

[Ctrl+L Ctrl+C Alt+Tab Ctrl+V RET Alt+Tab]

Ctrl+W

<N times>

Macro-Like

Ctrl+T ‘‘https://...’’ <N times>

Ctrl+F ‘‘free software’’

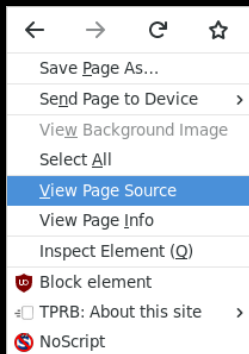
[Ctrl+L Ctrl+C Alt+Tab Ctrl+V RET Alt+Tab]

Ctrl+W

<N times>

- *Requires visual inspection* for conditional
- Still manual and tedious—what if there were 1000 URLs?

Lifting the Curtain



```
631 <section class="speaker-block" id="tp-speaker-12">
632 <div class="row"> <!-- speaker-12 row start -->
633 <!-- speaker-12 img column start -->
634 <div class="col-md-2 col-sm-3 col-xs-4">
635 
639 <header class="speaker-header" id="gerwitz">
640 <hgroup>
641 <h2>Mike Gerwitz</h2>
642 </hgroup>
643 </header>
644 <p><em>Computational symbiosis: Methods that meld mind and machine</em></p>
645 <p>Mike Gerwitz is a free software hacker and activist with a focus on
646 user privacy and security. He holds various volunteer roles within
647 GNU, including software evaluation and administrative duties. He has
648 twenty years of programming experience and his professional duties
649 range from Web development to compiler construction. He does nearly
650 all of his computing within the comfort of a terminal using
651 exclusively free software. Mike spends most of his free time with his
652 wife and two sons; he spends his remaining free time primarily on
653 hacking, research, volunteer work, and activism.</p>
654 <p><em>Photo taken by Kori Feener and courtesy of the Free Software Foundation, I
655 </div> <!-- speaker-12 content column end -->
```

free software Highlight All Match Case Whole Words 23 of 50

Text.

Text is a Universal Interface

The Shell Command Prompt

```
mikegerwitz@lp2019-laptop:~$
```

```
# ^ user      ^ host        ^ working directory (home)
```

The Shell Command Prompt

```
mikegerwitz@lp2019-laptop:~$  
# ^ user      ^ host          ^ working directory (home)
```

This presentation will show:

```
$ command  
output line 1  
output line 2  
...  
output line N
```

Eliminating the Web Browser

```
$ wget https://libreplanet.org/2019/speakers/
```

Eliminating the Web Browser

```
$ wget https://libreplanet.org/2019/speakers/

--2019-03-24 00:00:00-- https://libreplanet.org/2019/speakers/
Resolving libreplanet.org (libreplanet.org)... 209.51.188.248
Connecting to libreplanet.org (libreplanet.org)|209.51.188.248|
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

...
2019-03-24 00:00:00 (1.78 MB/s) - 'index.html' saved [67789]
```

Eliminating the Web Browser

```
$ wget https://libreplanet.org/2019/speakers/

--2019-03-24 00:00:00-- https://libreplanet.org/2019/speakers/
Resolving libreplanet.org (libreplanet.org)... 209.51.188.248
Connecting to libreplanet.org (libreplanet.org)|209.51.188.248:
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

...
2019-03-24 00:00:00 (1.78 MB/s) - 'index.html' saved [67789]

$ wget -O speakers.html \
    https://libreplanet.org/2019/speakers/
```

Browser vs. wget Comparison

```
Ctrl+L ‘‘https://libreplanet.org/2019/speakers/’’
```

```
$ wget https://libreplanet.org/2019/speakers/
```


Finding Text on the Command Line

```
$ grep 'free software' speakers.html
```

Finding Text on the Command Line

```
$ grep 'free software' speakers.html
```

```
⋮
```

```
<p>Mike Gerwitz is a free software hacker and activist with a f  
exclusively free software. Mike spends most of his free time wi
```

```
⋮
```

A More Gentle Reply

```
$ grep --quiet 'free software' speakers.html && echo yes
```

```
yes
```

A More Gentle Reply

```
$ grep --quiet 'free software' speakers.html && echo yes
```

```
yes
```

```
$ echo 'Hello, world!'
```

```
Hello, world!
```

A More Gentle Reply

```
$ grep --quiet 'free software' speakers.html && echo yes
```

```
yes
```

```
$ grep --quiet 'open source' speakers.html || echo no
```

```
no
```

Writing to Files (Redirection)

- Commands write to standard out (stdout) by default
- *Output redirection* writes somewhere else

```
# overwrites each time
```

```
$ echo 'Hello, world!' > hello.txt
```

```
$ echo 'Hello again, world!' > hello.txt
```

```
# appends (echo adds a newline)
```

```
$ echo 'First line' >> results.txt
```

```
$ echo 'Second line' >> results.txt
```

```
$ wget --quiet -O speakers.html \  
  https://libreplanet.org/2019/speakers/ \  
&& grep --quiet 'free software' speakers.html \  
  || echo https://libreplanet.org/2019/speakers/ \  
  >> results.txt
```

```
$ URL=https://libreplanet.org/2019/speakers/
$ wget --quiet -O speakers.html \
    "$URL" \
    && grep --quiet 'free software' speakers.html \
        || echo "$URL" \
            >> results.txt
```



```
$ URL=https://libreplanet.org/2019/speakers/
$ wget -qO speakers.html \
    "$URL" \
    && grep -q 'free software' speakers.html \
    || echo "$URL" \
    >> results.txt
```

```
$ URL=https://libreplanet.org/2019/speakers/
$ wget -qO - \
    "$URL" \
    | grep -q 'free software' \
    || echo "$URL" \
    >> results.txt
```

```
$ URL=https://libreplanet.org/2019/speakers/  
$ wget -qO - "$URL" \  
    | grep -q 'free software' || echo "$URL" >> results.txt
```

```
$ alias fetch-url='wget -q0-'
```

```
$ URL=https://libreplanet.org/2019/speakers/
```

```
$ fetch-url "$URL" \  
  | grep -q 'free software' || echo "$URL" >> results.txt
```

```
# original:

$ wget --quiet -O speakers.html \
    https://libreplanet.org/2019/speakers/ \
    && grep --quiet 'free software' speakers.html \
    || echo https://libreplanet.org/2019/speakers/ \
    >> results.txt

$ alias fetch-url='wget -qO-'

$ URL=https://libreplanet.org/2019/speakers/
$ fetch-url "$URL" \
    | grep -q 'free software' || echo "$URL" >> results.txt
```

Text is a Universal Interface

*“Expect the output of every
program to become the input
to another”*

—Doug McIlroy (1978)

The Unix Philosophy

This is the Unix philosophy: Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

—Doug McIlroy


```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
  | grep -A5 speaker-header \  
  | head -n14  
<header class="keynote-speaker-header" id="garbee">  
<hgroup>  
<h2>Bdale Garbee</h2>  
</hgroup>  
</header>  
<p><em>Closing keynote</em></p>  
--  
<header class="keynote-speaker-header" id="loubani">  
<hgroup>  
<h2>Tarek Loubani</h2>  
</hgroup>  
</header>  
<p><em>Opening keynote (Day 1)</em></p>  
--
```

```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
  | grep -A5 speaker-header \  
  | grep '<em>'
```

```
<p><em>Closing keynote</em></p>
```

```
<p><em>Opening keynote (Day 1)</em></p>
```

```
<p><em>Opening keynote (Day 2)</em></p>
```

```
[...]
```

```
<p><em>The Tor Project: State of the Onion</em> and <em>Library</em></p>
```

```
<p><em>The Tor Project: State of the Onion</em></p>
```

```
[...]
```

```
<p><em>Large-scale collaboration with free software</em></p>
```

```
<p><em>Large-scale collaboration with free software</em></p>
```

```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
  | grep -A5 speaker-header \  
  | grep -o '<em>[^<]\+</em>'  
<em>Closing keynote</em>  
<em>Opening keynote (Day 1)</em>  
<em>Opening keynote (Day 2)</em>  
[...]  
<em>The Tor Project: State of the Onion</em>  
<em>Library Freedom Institute: A new hope</em>  
<em>The Tor Project: State of the Onion</em>  
[...]  
<em>Large-scale collaboration with free software</em>  
<em>Large-scale collaboration with free software</em>
```

```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
| grep -A5 speaker-header \  
| grep -o '<em>[^<]\+</em>' \  
| sort \  
| uniq -cd  
2 <em>Hackerspace Rancho Electrónico</em>  
4 <em>Large-scale collaboration with free software</em>  
2 <em>Library Freedom Institute: A new hope</em>  
2 <em>Right to Repair and the DMCA</em>  
2 <em>Teaching privacy and security via free software</em>  
2 <em>The joy of bug reporting</em>  
5 <em>The Tor Project: State of the Onion</em>
```

```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
| grep -A5 speaker-header \  
| grep -o '<em>[^<]\+</em>\' \  
| sort \  
| uniq -cd \  
| sort -nr \  
| head -n5  
5 <em>The Tor Project: State of the Onion</em>  
4 <em>Large-scale collaboration with free software</em>  
2 <em>The joy of bug reporting</em>  
2 <em>Teaching privacy and security via free software</em>  
2 <em>Right to Repair and the DMCA</em>
```

```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
  | grep -A5 speaker-header \  
  | grep -o '<em>[^<]+\</em>' \  
  | sort \  
  | uniq -cd \  
  | sort -nr \  
  | head -n5 \  
  | sed 's#^ *\(.\+\) <em>\(.*\)\</em>#\2 has \1 speakers#'  
The Tor Project: State of the Onion has 5 speakers  
Large-scale collaboration with free software has 4 speakers  
The joy of bug reporting has 2 speakers  
Teaching privacy and security via free software has 2 speakers  
Right to Repair and the DMCA has 2 speakers
```

```
$ fetch-url https://libreplanet.org/2019/speakers/ \  
| grep -A5 speaker-header \  
| grep -o '<em>[^<]+\</em>' \  
| sort \  
| uniq -cd \  
| sort -nr \  
| head -n5 \  
| sed 's#^ *\(..\+\) <em>\(.*\)\</em>#\2 has \1 speakers#'  
| espeak
```

Incremental Development

Interactive REPL, Iterative Decomposition


```
$ grep -o 'https\?://[^\ ]\+' email-of-links.txt
https://en.wikipedia.org/wiki/Free_software
https://en.wikipedia.org/wiki/Open_source
https://en.wikipedia.org/wiki/Microsoft
https://opensource.org/about
```

```
$ grep -o 'https\?://[^\ ]\+' email-of-links.txt \  
  | while read URL; do  
    echo "URL is $URL"  
  done  
URL is https://en.wikipedia.org/wiki/Free_software  
URL is https://en.wikipedia.org/wiki/Open_source  
URL is https://en.wikipedia.org/wiki/Microsoft  
URL is https://opensource.org/about
```

```
$ grep -o 'https\?://[^\ ]\+' email-of-links.txt \  
  | while read URL; do \  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL \  
  done \  
> results.txt
```

```
$ grep -o 'https\?://[^\ ]\+' email-of-links.txt \  
  | while read URL; do \  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL" \  
  done \  
  | tee results.txt  
https://en.wikipedia.org/wiki/Microsoft  
https://opensource.org/about
```

```
$ grep -o 'https\?://[^\ ]\+' email-of-links.txt \  
| while read URL; do \  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL" \  
done \  
| tee results.txt  
| xclip -i -selection clipboard
```

```
$ grep -o 'https\?://[^\ ]\+' email-of-links.txt \  
  | while read URL; do \  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL" \  
  done \  
  | tee >( xclip -i -selection clipboard )  
https://en.wikipedia.org/wiki/Microsoft  
https://opensource.org/about
```

```
$ xclip -o -selection clipboard \  
  | grep -o 'https\?://[~ ]\+' \  
  | while read URL; do \  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL" \  
  done \  
  | tee >( xclip -i -selection clipboard )  
https://en.wikipedia.org/wiki/Microsoft  
https://opensource.org/about
```

```
$ xclip -o -selection clipboard \  
  | grep -o 'https\?://[^\ ]\+' \  
  | while read URL; do \  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL" \  
  done \  
  | tee results.txt  
https://en.wikipedia.org/wiki/Microsoft  
https://opensource.org/about  
  
$ xclip -i -selection clipboard < results.txt
```


Go Grab a Coffee

```
$ sleep 1 && echo done & echo start  
start  
done
```

```
$ sleep 1 && echo done & echo start  
start  
done
```

(Don't do this for large numbers of URLs!)

```
$ while read URL; do  
    fetch-url "$URL" | grep -q 'free software' \  
    || echo "$URL" &  
done | tee results.txt
```

```
url-grep
```

```
#!/bin/bash
```

```
search=$1
```

```
url=$2
```

```
wget -q0- "$url" \  
  | grep -q "$search" || echo "$url"
```

```
$ chmod +x url-grep
```

```
$ while read URL; do  
  ./url-grep 'free software' "$URL" >> results.txt  
done
```

```
url-grep
```

```
#!/bin/bash
```

```
search=$1
```

```
url=$2
```

```
wget -q0- "$url" \  
  | grep -q "$search" || echo "$url"
```

```
$ chmod +x url-grep
```

```
$ xargs -n1 ./url-grep 'free software' > results.txt
```

```
url-grep
```

```
#!/bin/bash
```

```
search=$1
```

```
url=$2
```

```
wget -q0- "$url" \  
  | grep -q "$search" || echo "$url"
```

```
$ chmod +x url-grep
```

```
$ xargs -n1 -P5 ./url-grep 'free software' > results.txt
```

```
#           ^ 5 concurrent processes
```

Execution Time

```
$ wc -l url-list  
1000
```

```
$ time xargs -n1 -P10 ./url-grep 'free software' < url-list  
real    0m17.548s  
user    0m8.283s  
sys     0m4.877s
```

Resize Images

```
$ for img in *.png; do
    convert "$img" -resize 50% "sm-$img"
done

# nested directories
$ find . -name '*.png' -exec convert {} -resize 50% sm-{} \;
```


Password Generation

```
# generate a random 32-character password
$ tr -cd '[:graph:]' < /dev/urandom | head -c32
'TB~cmJQ1%S8&tJ,%FoD54}"Fm4}\Iwi

# generate passphrase from EFF's large dice wordlist
# (https://www.eff.org/dice)
$ cut -f2 eff_large_wordlist.txt \
  | sort -R --random-source=/dev/urandom \
  | head -n6 \
  | tr '\n' ' '
oppressor roman jigsaw unhappy thinning grievance
```

Password Manager

```
$ gpg --decrypt password-db.gpg | head -n3  
https://foo.com  
user mikegerwitz  
pass !({:pT6DcJG.cr&0Aco_EC63r_*xg|uD
```

Password Manager

```
$ gpg --decrypt password-db.gpg | head -n3
https://foo.com
user mikegerwitz
pass !({:pT6DcJG.cr&0Aco_EC63r_*xg|uD
```

```
$ gpg --decrypt password-db.gpg \
  | grep -A2 ^https://foo.com \
  | tail -n2 \
  | while read key value; do
    echo "paste $key..."
    printf %s "$value" \
      | xclip -o -selection clipboard -l 1 -quiet
  done
```

Remote Password Manager With 2FA

- Add `extra-socket` to `.gnupg/gpg-agent.conf`
- Add `RemoteForward` in `.ssh/config` for host
- Save script on previous slide as `get-passwd`

```
$ ssh -Y mikegerwitz-pc get-passwd https://foo.com
```

Taking Screenshots

```
# draw region on screen, output to ss.png
$ import ss.png

# screenshot of entire screen after 5 seconds
$ import -pause 5 -window root ss.png

# screenshot to clipboard
import png:- | xclip -i -selection clipboard -t image/png
```

Screenshot OCR

```
$ import png:- | tesseract -psm 3 - - | cowsay
```

```
-----  
/ Keynote Speakers \  
|                    |  
| i                  |  
|                    |  
| Bdale Garbee Micky Metts Richard |  
\ Stallman Tarek Loubani /  
-----
```

```
 \  ^__^  
  \ (oo)\_____   
    (__)\       )\/\  
       ||----w |  
       ||     ||
```

Screenshot OCR

```
# OCR at URL
$ wget -qO- https://.../keynote_banner.png \
  | tesseract -psm 3 - -

# perform OCR on selected region and speak it
$ import png:- | tesseract -psm 3 - - | espeak

# perform OCR on clipboard image and show in dialog
$ xclip -o -selection clipboard -t image/png \
  | tesseract -psm 3 - - \
  | xargs -0 zenity --info --text
```

```
$ xdotool search --name ' GNU IceCat$' windowactivate --sync \  
    window size 1024 600 \  
    key ctrl+t \  
&& while read url; do  
    xdotool getactivewindow \  
        key ctrl+l type "$url" \  
    && xdotool getactivewindow \  
        key Return sleep 5 \  
        key ctrl+f sleep 0.5 \  
        type 'free software' \  
    && import -window "$( xdotool getactivewindow )" \  
        -crop 125x20+570+570 png:- \  
        | tesseract -psm 3 - - \  
        | grep -q ^Phrase && echo "$url" \  
done < url-list | tee results.txt
```


Getting Help

- All GNU packages have Info manuals and `--help`
- Most programs (including GNU) have manpages

```
$ grep --help      # usage information for grep
```

```
$ man grep        # manpage for grep
```

```
$ info grep       # full manual for grep
```

```
# bash help
```

```
$ help
```

```
$ man bash
```

Mike Gerwitz
mtg@gnu.org

Slides and Source Code Available Online
<<https://mikegerwitz.com/talks/cs4m.pdf>>

Licensed under the Creative Commons Attribution ShareAlike 4.0
International License